305-CD-028-002

EOSDIS Core System Project

Release B CSMS Communications Subsystem Design Specification for the ECS Project

March 1996

Hughes Information Technology Systems
Upper Marlboro, Maryland

Release B CSMS Communications Subsystem Design Specification for the ECS Project

March 1996

Prepared Under Contract NAS5-60000
CDRL Item 046

SUBMITTED BY

Rick Kochhar /s/	3/29/96
Rick Kochhar, Release B CCB Chairman EOSDIS Core System Project	Date

Hughes Information Technology Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Preface

This document is one of nineteen comprising the detailed design specifications of the SDPS and CSMS subsystem for Release B of the ECS project. A complete list of the design specification documents is given below. Of particular interest are documents number 305-CD-020, which provides an overview of the subsystems and 305-CD-039, the Data Dictionary, for those reviewing the object models in detail.

The SDPS and CSMS subsystem design specification documents for Release B of the ECS Project include:

305-CD-020	Release B Overview of the SDPS and CSMS Segment System Design Specification
305-CD-021	Release B SDPS Client Subsystem Design Specification
305-CD-022	Release B SDPS Interoperability Subsystem Design Specification
305-CD-023	Release B SDPS Data Management Subsystem Design Specification
305-CD-024	Release B SDPS Data Server Subsystem Design Specification
305-CD-025	Release B SDPS Ingest Subsystem Design Specification
305-CD-026	Release B SDPS Planning Subsystem Design Specification
305-CD-027	Release B SDPS Data Processing Subsystem Design Specification
305-CD-028	Release B CSMS Segment Communications Subsystem Design Specification
305-CD-029	Release B CSMS Segment Systems Management Subsystem Design Specification
305-CD-030	Release B GSFC Distributed Active Archive Center Design Specification
305-CD-031	Release B LaRC Distributed Active Archive Center Design Specification
305-CD-033	Release B EDC Distributed Active Archive Center Design Specification
305-CD-034	Release B ASF Data Center Distributed Active Archive Center Design Specification
305-CD-035	Release B NSIDC Distributed Active Archive Center Design Specification
305-CD-036	Release B JPL Distributed Active Archive Center Design Specification
305-CD-037	Release B ORNL Distributed Active Archive Center Design Specification
305-CD-038	Release B System Monitoring and Coordination Center Design Specification
305-CD-039	Release B Data Dictionary for Subsystem Design Specification

Object models presented in this document have been exported directly from CASE or DBMS tools and in some cases contain too much detail to be easily readable within hard copy page constraints. The reader is encouraged to view these drawings on line using the Portable Document Format (PDF) electronic copy available via the ECS Data Handling System (EDHS) at: URL http://edhs1.gsfc.nasa.gov.

This document is a formal contract deliverable with an approval code of 2; as such it requires Government review and approval prior to acceptance and use. This document is under ECS contractor configuration control. Once this document is approved, Contractor approved changes are handled in accordance with Class I and Class II change control requirements described in the EOS Configuration Management Plan, and changes to this document shall be made by document change notice (DCN) or by complete revision.

Any questions or proposed changes should be addressed to:

Data Management Office
The ECS Project Office
Hughes Information Technology Systems
1616 McCormick Drive
Upper Marlboro, MD 20774-5372

Abstract

The Communications Subsystem (CSS) provides the overall communications infrastructure, and the communications services to support other subsystems in the Science and Communications Development Office (SCDO) and the Flight Operations Segment (FOS). This document provides the design at Release B of the communications infrastructure and services provided by CSS.

Keywords: CSMS, CSS, Communications, DCE, OODCE, Release B, Time, Directory, Naming, Security, Message Passing, Distributed Objects, LifeCycle, Threads, Multicast, GSS, Virtual Terminal, Bulk Data Service, File Transfer, Bulletin Board, Mail, Event Logging, Process Framework, Server Request Framework, Subscription Service, Universal Reference.

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Submitted As Final
iii through xiv	Submitted As Final
1-1 and 1-2	Submitted As Final
2-1 through 2-4	Submitted As Final
3-1 through 3-18	Submitted As Final
4-1 through 4-416	Submitted As Final
5-1 through 5-4	Submitted As Final
A-1 through A-36	Submitted As Final
AB-1 through AB-6	Submitted As Final

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
305-CD-028-001	Preliminary	October 1995	95-0763
305-CD-028-002	Submitted As Final	March 1996	96-0233

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

2. Related Documentation

3. Communications Subsystem Overview

4. Communications Subsystem (CSS) Design

5. CSS HWCI

Figures

Tables

Appendix A. Requirements Trace

Abbreviations and Acronyms

This page intentionally left blank.

1. Introduction

1.1 Identification

The Release B CSMS Communications Subsystem Design Specification for the ECS Project, Contract Data Requirement List (CDRL) item 046, with requirements specified in Data Item Description (DID) 305/DV2, is a required deliverable under the Earth Observing System Data and Information System (EOSDIS) Core System (ECS), Contract NAS5-60000. This publication is part of a series of documents comprising the Science and Communications Development Office design specification for the Communications and System Management segment (CSMS) and the Science and Data Processing Subsystem (SDPS) for Release B.

1.2 Scope

The Release B CSMS Communications Subsystem Design Specification defines the design. It defines the Communications Subsystem computer software and hardware architectural design, as well as subsystem design based on Level 4 requirements.

This document reflects the February 14, 1996 Technical Baseline maintained by the ECS contractor configuration control board in accordance with ECS Technical Direction No.11, dated December 6, 1994.

1.3 Document Organization

The document is organized to describe the Release B CSMS Communications Subsystem design as follows:

Section 1 provides information regarding the identification, scope, status, and organization of this document.

Section 2 provides a listing of the related documents, which were used as source information for this document.

Section 3 provides an overview of the Subsystem, focusing on the high-level design concept. This provides general background information to put the Communications subsystem into context.

Sections 4 contains the structure of the computer software configuration item (CSCI) comprising the Communications Subsystem.

Section 5 contains the hardware configuration item (HWCI) design of the Communications Subsystem.

The section Abbreviations and Acronyms contains an alphabetized list of the definitions for abbreviations and acronyms used in this volume.

1.4 Status and Schedule

This submittal of DID 305/DV2 meets the milestone specified in the Contract Data Requirements List (CDRL) of NASA contract NAS5-60000. A previous version of this submittal was reviewed during the CSMS Release A Critical Design Review (CDR).

This page intentionally left blank.

2. Related Documentation

2.1 Parent Documents

The parent document is the document from which the scope and content of this Communications Subsystem Design Specification are derived.

194-207-SE1-001	System Design Specification for the ECS Project

2.2 Applicable Documents

The following documents are referenced within this Communications Subsystem Design Specification, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

209-CD-001-002	Interface Control Document Between EOSDIS Core System (ECS) and the NASA Science Internet (NSI)
209-CD-002-003	Interface Control Document Between EOSDIS Core System (ECS) and ASTER Ground Data System
209-CD-003-003	Interface Control Document Between EOSDIS Core System (ECS) and EOS-AM Project for AM-1 Spacecraft Analysis Software
209-CD-004-003	Data Format Control Document for the Earth Observing System (EOS) AM-1 Project Data Base
209-CD-005-005	Interface Control Document Between EOSDIS Core System (ECS) and Science Computing Facilities (SCF)
209-CD-006-005	Interface Control Document Between EOSDIS Core System (ECS) and National Oceanic and Atmospheric Administration (NOAA) Affiliated Data Center (ADC)
209-CD-007-003	Interface Control Document Between EOSDIS Core System (ECS) and TRMM Science Data and Information System (TSDIS)
209-CD-008-004	Interface Control Document Between EOSDIS Core System (ECS) and the Goddard Space Flight Center (GSFC) Distributed Active Archive Center (DAAC)
209-CD-009-002	Interface Control Document Between EOSDIS Core System (ECS) and the Marshall Space Flight Center (MSFC) Distributed Active Archive Center (DAAC)
209-CD-011-004	Interface Control Document Between EOSDIS Core System (ECS) and the Version 0 System
305-CD-012-001	Release A CSMS Communications Subsystem Design Specification for the ECS Project
308-CD-001-005	Software Development Plan for the ECS Project

313-CD-006-002	Release B SDPS/CSMS Internal Interface Control Document for the ECS Project
423-41-03	Goddard Space Flight Center, EOSDIS Core System (ECS) Contract Data Requirements Document
none	Open Software Foundation, OSF DCE 1.0.3 Application Development Guide

2.3 Information Documents Not Referenced

The following documents, although not referenced herein and/or not directly applicable, do amplify and clarify the information presented in this document. These documents are not binding on the content of this CSMS Subsystem Design Specifications.

205-CD-002-002	Science User's Guide and Operations Procedure Handbook for the ECS Project. Part 4: Software Developer's Guide to Preparation, Delivery, Integration, and Test with ECS
206-CD-001-002	Version 0 Analysis Report for the ECS Project
209-CD-010-002	Interface Control Document Between EOSDIS Core System (ECS) and the Langley Research Center (LaRC) Distributed Active Archive Center (DAAC)
302-CD-003-001	Release B Facilities Plan for the ECS Project
101-303-DV1-001	Individual Facility Requirements for the ECS Project, Preliminary
194-317-DV1-001	Prototyping and Studies Plan for the ECS Project
318-CD-000-XXX	Prototyping and Studies Progress Report for the ECS Project (monthly)
333-CD-003-002	SDP Toolkit Users Guide for the ECS Project
601-CD-001-004	Maintenance and Operations Management Plan for the ECS Project
604-CD-001-004	Operations Concept for the ECS Project: Part 1-- ECS Overview
604-CD-002-003	Operations Concept for the ECS project: Part 2B -- ECS Release B
604-CD-003-002	Operations Concept for the ECS Project: Part 2A -- ECS Release A
604-CD-004-001	Operations Concept for the ECS Project: Part 2 -- FOS
101-620-OP2-001	List of Recommended Maintenance Equipment for the ECS Project
194-703-PP1-001	System Design Review (SDR) Presentation Package for the ECS Project
194-813-SI4-002	Planning and Scheduling Prototype Results Report for the ECS Project
194-813-SI4-003	DADS Prototype One FSMS Product Operational Evaluation for the ECS Project
194-813-SI4-004	DADS Prototype One STK Wolfcreek 9360 Automated Cartridge System Hardware Characterization Report for the ECS Project
813-RD-009-001	DADS Prototype Two Multi-FSMS Product Integration Evaluation for the ECS Project
828-RD-001-002	Government Furnished Property for the ECS Project

193-TP-626-001	GCDIS/UserDIS Study for the ECS Project, Technical Paper
193-WP-118-001	Algorithm Integration and Test Issues for the ECS Project
193-WP-611-001	Science-based System Architecture Drivers for the ECS Project, Revision 1.0
193-WP-623-001	ECS Evolutionary Development White Paper
194-TP-266-002	Data Distribution Architecture Logical Object Model (LOM) for the ECS Project, Version 2.01
194-TP-267-001	Data Server Architecture Logical Object Model (LOM) for the ECS Project, Version 2.00
194-TP-313-001	ECS User Characterization Methodology and Results for the ECS Project
194-TP-316-002	Data Compression Study for the ECS Project
194-TP-548-001	User Scenario Functional Analysis [for the ECS Project]
194-TP-569-001	PDPS Prototyping at ECS Science and Technology Laboratory, Progress Report #4
194-WP-901-002	EOSDIS Core System Science Information Architecture, White Paper, Working Paper
194-WP-902-002	ECS Science Requirements Summary, White Paper, Working Paper
194-WP-904-002	Multi-Track Development for the ECS Project, White Paper, Working Paper
194-WP-913-003	User Environment Definition for the ECS Project, White Paper, Working Paper
194-WP-914-001	CORBA Object Request Broker Survey for the ECS Project, White Paper, Working Paper
194-WP-918-001	DADS Prototype One FSMS Product Operational Evaluation, White Paper, Draft Report
194-WP-925-001	Science Software Integration and Test, White Paper for the ECS Project
222-TP-003-008	Release Plan Content Description for the ECS Project
410-TD-001-002	ECS User Interface Style Guide, Technical Data for the ECS Project
420-WP-001-001	Maximizing the Use of COTS Software in the SDPS SDS Software Design, White Paper for the ECS Project
430-TP-001-001	SDP Toolkit Implementation with Pathfinder SSM/I Precipitation Rate Algorithm, Technical Paper for the ECS Project
423-16-01	Goddard Space Flight Center, Data Production Software and Science Computing Facility (SCF) Standards and Guidelines
423-16-02	Goddard Space Flight Center, PGS Toolkit Requirements Specification for the ECS Project

423-41-02	Goddard Space Flight Center, Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System (ECS)
540-022	Goddard Space Flight Center, Earth Observing System (EOS) Communications (Ecom) System Design Specification
560-EDOS-0211.0001	Goddard Space Flight Center, Interface Requirements Document Between EDOS and the EOS Ground System (EGS)

3. Communications Subsystem Overview

3.1 Introduction

The Communications Subsystem (CSS) provides for the interconnection of users and service providers, transfer of information within the Earth Observing System Data Information System (EOSDIS) Core System (ECS) and between ECS and many EOSDIS components, and management of all ECS communications components. It supports and interacts with the Management Subsystem (MSS), Flight Operations Segment (FOS) and the other ECS subsystems. This section provides the following overview and critical design characterization for CSS:

- **Release Requirements and Service Deployment**

  Characterizes the cumulative functionality required of CSS within the Interim Release 1 (Ir1), Release A (RA) missions and Release B (RB)
- **Subsystem Overview**

  Provides a first-level decomposition of CSS into its four service groupings
- **CSS Context**

  Establishes the context of CSS within the ECS/EOSDIS Release B environment

3.2 Requirements

A complete description of CSS requirements is provided in Section 4 of this document.

3.2.1 Interim Release 1

For Ir1, traditional TCP/IP application services need to be provided by CSS. These include ftp and telnet services which Science Computing Facilities (SCFs) need to upload TRMM and Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument algorithm packages and interact with integration and test (I&T) hosts at the 4 Ir1 sites (EDC, Langley Research Center (LaRC), Marshall Space Flight Center (MSFC) and Goddard Space Flight Center (GSFC)). CSS security and time (OSF Distributed Time Service) services are also expected to be used for the limited TRMM ingest interface testing planned for Ir1. The OSF Distributed Computing Environment (DCE) Cell Directory Service (CDS) and Remote Procedure Call (RPC) services will also be available for MSS, PDPS, Data Servers, CIDM, INS developers for use in ingest testing and algorithm I&T applications.

3.2.2 Release A

For Release A, additional services are needed from CSS. These include secure implementations of ftp and telnet services, electronic mail (e-mail) exchange with external systems and bulletin board capabilities. The CSS object-encapsulated services will also require additional functionality in support of ECS subsystems application event collection and processing, interprocess communications, and thread services. The CSS Distributed Object Framework will be provided to encapsulate all CSS services requiring object-oriented, client-server or peer-peer execution.

3.2.3 Release B

For Release B, additional services are provided by CSS to support the AM-1, Landsat-7, Metor, Adeos11, Radar, ALT, FOO missions. The CSS object-encapsulated services will also provide additional functionality in support of ECS subsystems application event collection and processing, interprocess communications, and thread services. The CSS Distributed Object Framework will be provided to encapsulate all CSS services requiring object-oriented, client-server or peer-peer execution. The CSS common facility services include enhanced implementation of Remote File Access, electronic mail (e-mail) exchange with external systems, and bulletin board capabilities. The CSS services also include new infrastructure services Managed Process Framework (MPF), Server Request Framework (SRF), Universal Reference (UR) & subscription as a part of Rel B.

3.3 Communications Subsystem Overview

CSS provides a basket of infrastructural services, characterized as “middleware,” for use in all the subsystems (MSS, DSS, DPS, FOS, CLS, IOS, DMS, PLS, INS) of ECS. This infrastructure consists of communication services that application developers will use in the development of distributed applications, which enables these applications to interact with other applications within and outside of ECS. These services are standards-based and are interoperable (hardware and vendor independent). These services are broadly classified into four categories:

- Common Facilities
- Object Services
- Distributed Object Framework (DOF).
- Infratructure Services.

The Common Facilities include legacy communications services required within the ECS infrastructure for file transfer, electronic mail, bulletin board and remote terminal support. The Object Services support all ECS applications with interprocess communication and specialized infrastructural services such as security, directory, time, asynchronous message passing, event logging, lifecycle. The Distributed Object Framework is a collection of a set of core object services, collectively providing object-oriented client server development and interaction amongst applications.

Interim Release 1 provides ftp, virtual terminal and DCE core services: Directory, Security, Time, RPCs. Release A provides mail, bulletin board, event logger, Message Passing and object oriented DCE services along with some enhancements. Both Ir1 and Release A use a single DCE cell where all the users, platforms, and services are maintained and belong to.

In Release B timeframe, ECS will be implementing an enhanced "Version 1" protocol which provides EOSDIS users with access to an extended set of information services. The ECS gateway will provide the translation of the V0 protocols for user queries and query responses to the equivalent ECS "Version 1" protocol. Please note that through the gateway, users will be limited to the suite of services provided by the Version 0 system.

As part of the gateway security effort, we will be supporting access from Kerberized clients (that are using sockets) to the ECS gateway. This will allow the ECS system to more securely authenticate the externals who use gateway services.

Release B will use multiple DCE cells with additional functionalities: Enhanced (DCE 1.1) security, & lifecycle.

3.3.1 Physical Subsystem Context

CSS is distributed across all ECS components. On client and server platforms, the CSS provides, ECS subsystems applications with access to legacy services such as mail, bulletin board, file transfer and host access as well as object-oriented infrastructure services upon which to execute client-server operations. Client platforms outside the ECS installation are provided with a subset of CSS services which are integrated within the ECS Toolkit software. In addition to installation on ECS subsystems platforms, CSS services are also installed on CSS and MSS servers and workstations distributed throughout the ECS. Further detail regarding the physical hardware and software composition, sizing and siting of CSS is provided in later sections of this document.

3.3.2 Context Description

Figure 3.3.2-1 illustrates the Release B interface context for CSS. Through its distributed communications service infrastructure, CSS supports the Tropical Rainforest Measuring Mission (TRMM) mission, AM-1 mission, Landsat-7, Metor, Adeos11, Radar ALT FOO, ECS operations and maintenance management, and EOSDIS user access. CSS will also support access to V0 (V0 interoperability) and to V0 migrated data.

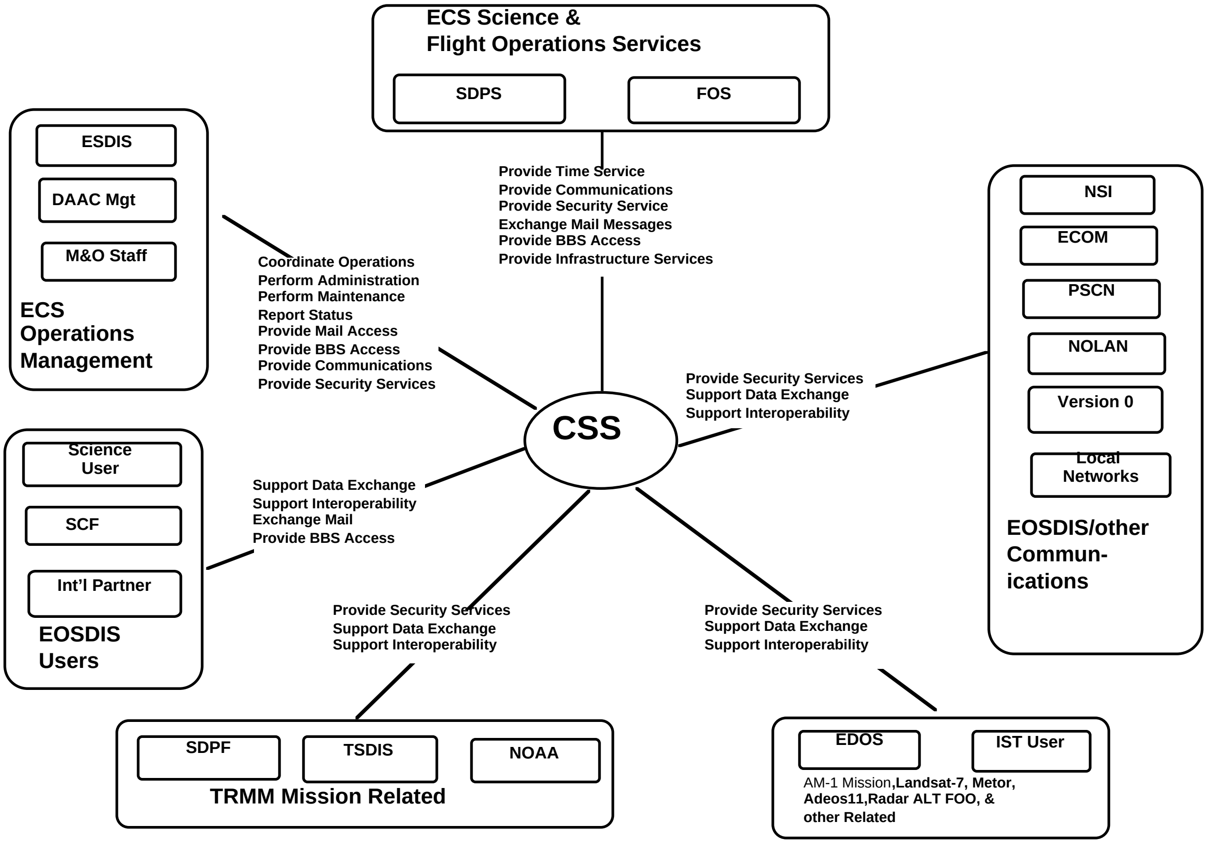


Figure 3.3.2-1. CSS Context (Release B)

Through the NASA Science EBNet and local institutional network connectivity, CSS supports Science Data Processing service access by ECS user communities, including TRMM scientists and AM-1. These interfaces facilitate exchange of various Release B data items (e.g., science products, algorithms, ancillary data sets) as well as interactive access sessions. Within ECS, communications services are provided to support ECS subsystems applications and infrastructure to facilitate peer-to-peer, client-server communication including required value-added services.

3.4 Subsystem CSCI and HWCI Structure

The Communications Subsystem is composed of one software CSCI, Distributed Communication Software CI (CSS-DCSCI); and one hardware CSCI, Distributed Communications Hardware CI (CSS-DCHCI).

3.4.1 CSCI Summary

This software CSCI consists of a number of services; a brief summary of each of the CSCIs is described here. A more detailed version of these services is presented in Section 4 of this document.

Table 3.4.1-1 CSS Physical Design Mapping (1 of 4)

CSC	Class
DOF	DCEInterface
	DCEInterfaceMgr
	DCEObj
	DOF
	ESO
	Naming
	Security
Universal Reference	EcUrClassID
	EcUrUR
	EcUrURMaker
	EcUrURProvider
	EcUrURProviderMaker
	GrLiAnyReferencedClass
	GrLiAnyURClass
Event Logger	EcUtLoggerRelA
	EcUtLoggerRelAAudit
	EcUtLoggerRelADebug
	EcUtLoggerRelAFault
	EcUtLoggerRelAMgmt
	EcUtLoggerRelAPerf
	EcUtLoggerRelASec
	strstream
Electronic Mail & Bulletin Board	CsBBMailRelA
	CsEmMailRelA

Table 3.4.1-1 CSS Physical Design Mapping (2 of 4)

CSC	Class
Directory/Naming	CsDcXds
	DNS
	Directory_Naming_Service
	EcDnAttribute
	EcDnCompositeName
	EcDnContext
	EcDnElement
	EcDnValue
	X.500
LifeCycle	DCEActivation
Message Passing	EcMpMsgCb
	EcMpMsgPsngCtrl
	EcMpQueue
	EcMpQueueCbIn
	EcMpQueueIn
	EcMpQueueOut
	EcMpSessionList
	EcMpTransferCli
	EcMpTransferSrv
	Message_Passing_Service
	RWFile
	RWPtrDlist
	EcDcDSyncCom
Thread	Pthread
	PthreadCond
	PthreadInterval
	PthreadMutex
	PthreadTime
	ThisPthread
FTP	CsFtFTPRelB
Subscription Service	EcClAction
	EcClCollectorVector
	EcClEvent
	EcClGenConnector
	EcClSubscription
	EcClSubscriptionCollector
	EcShActionBase
	EcShSubscription
	EcClAction

Table 3.4.1-1 CSS Physical Design Mapping (3 of 4)

CSC	Class
	EcClEvent
	EcClSubscription
	EcSbAction
	EcSbEvent
	EcSbSubscription
	EcShAction
	EcShEvent
	EcDcDSync
	EcDcDSyncCom
	EcSbNotification
	EcDbEventStore
	EcDbSubscriptionStore
	EcSbEventHandler
	EcSbSubscriptionHandler
	EcSbSubscriptionServer
	EcSbTimeKeeper
	EcSrGenConnector
Generic Security Service	EcSeGSSB
	EcSeGSSTCPB
Process Framework	AsyncMsg
	EcPfClient
	EcPfConfigFile
	EcPfGenProcess
	EcPfGenServer
	EcPfManagedServer
	EcPfUnmanagedServer
	GSO
	MSS_Stuff
	MyClientProc
	MyManagedServerProc
	MyUnmanagedServerProc
	ObjectLinkList
	SRF
Security Service	DCEAclMgr
	DCEAclSchema
	DCEInterface
	DCEInterfaceMgr
	DCEObj
	DCEPassword

Table 3.4.1-1 CSS Physical Design Mapping (4 of 4)

CSC	Class
	DCERefMon
	DCESecId
	DCEUuid
	ECSAcl
	ECSAclDb
	ECSAclStorageManager
	ECSModifyableAcl
	ESO
	EcSeSecurity
	EcSeServerKeyMgmt
	acl_edit
	appClientObj
	appServerObj
	app_1_0_ABS
	rgy_edit
Server Request Framework	AppAsynchRequest_C
	AppAsynchRequest_S
	AppRequestServer_C
	AppRequestServer_S
	EcMhMsgReceiver
	EcSrAsynchRequest_C
	EcSrAsynchRequest_S
	EcSrRequestDispatcher
	EcSrRequestServer_C
	EcSrRequestServer_S
	EcMhMsgEnvelope
	EcMhMsgHandler
	EcMhMsgReceiver
	EcMhPendingMsg
	EcMhXMessageHandler
	EcMpMsgCb
	EcMpSessionList
	EcUtStreamable
Time	EcFosTimeProviderB
	EcTiTimeService

Directory Naming Service

The Directory Naming Service provides a reliable mechanism by which distributed applications can associate information with names. Its primary purpose is to allow clients to locate servers. Its capabilities, however, are general-purpose, and it can be used in any application that needs to make names and their attributes available throughout a network.

CSS will provide implementation of both the DNS and the X.500 by supporting BIND and OSF Global Directory Service and OSF Cell Directory Service (CDS). It also provides application programmers the ability to store, retrieve, list information in the locally supported namespaces. The DNS and X.500 namespaces are used to connect the locally supported CDS namespaces. The functionality provided here will be implemented on top of XDS/XOM interfaces. As such, application programmers can use the above mentioned services (store, retrieve, list) in CDS as well as OSF GDS.

Security Service

The security service provides secure transfer of data on local and wide area networks. It provides mechanisms to verify the identity of users, and to determine whether users are permitted to invoke certain operations (authentication and authorization). Transmission of data may be protected through the use of checksums and encryption of data. Authentication is provided by trusted third party (secret key) authentication. Authorization is based on Access Control Lists. The protocol used for authentication is Kerberos. All of these features are implemented within the ECS domain by employing OSF/DCE Security Services.

Multicast

Multicasting is a mechanism through which a single copy of data is transferred from a single point to several places. Multicasting allows a sending application to specify a multicast address and send one copy of the data to that address. This data is then distributed through the Multicast backbone to all the applications listening at that address. This reduces the network traffic and improves the performance.

Multicast is being used by FOS as a Release B service. Some FOS testing with multicast service is expected to take place before Release B delivery.

Message Passing Service

The Message Passing Service allows for the exchange of information between applications running on different platforms. Clients send data to servers, which process the data and return the result back to the client. This interaction can be classified into three categories: synchronous, asynchronous, and deferred synchronous.

CSS will provide two implementations of Message Passing. The first model will provide for asynchronous and synchronous message passing—byte streams only—with store and forward, recovery and persistence. It will also include the concept of groups where a list of receivers belong to a group. A message sent to the group will be delivered to all the addresses registered in that local group. The second model will provide for asynchronous and deferred synchronous communication without recovery.

Both implementations are designed to take advantage of OODCE-provided DCE-Pthread class which is used to start and control the execution of a thread. The second mode requires more programmer involvement than the first model. Message Passing Service is generally intended to handle low volumes of data per message. Compare with k/ftp and DFS (below) for bulk data transfer.

Thread

A thread is a light weight process without the actual process overhead. Threads provide an efficient and portable way to provide asynchronous and concurrent processing, which is a requirement of network software. Threads can maintain thread specific data and can also share data with other threads in an application. This service provides functionality to create, maintain (scheduling, locking, etc.) threads.

Time

The Time Service keeps system (host) clocks in the ECS network approximately in sync by adjusting the time kept by the operating system at every host. This service changes the clock tick increments (rather than the actual clock) so that host clocks will be in sync with some reference time provided by an external time provider. CSS will also provide a way to simulate time by applying a supplied delta time to the actual time. Within ECS, OSF DTS will be used to sync the system clocks.

LifeCycle

Managing a system involves managing individual applications. An operator may want to start a new application, shutdown/suspend a running application due to anomalies. An application may not be active all the time to accept requests. In order to effectively use the CPU and memory it is desired to control the applications as well s some objects residing in the application by starting them on demand.

LifeCycle services can be broadly classified into two categories: Application and Object level. LifeCycle services for applications involve Startup, Shutdown, Suspend and Resume functionality on applications. This functionality lets the M&O manage server applications. MSS provides the application related LifeCycle functionality. CSS provides the internal APIs that are needed for the MSS to control the applications. LifeCycle services for objects provide the application programmer with the functionality to create and delete server objects residing in different address spaces. This functionality is currently embeded in MPF.

Distributed Object Framework (DOF)

In an object oriented processing architecture, objects may be distributed in multiple address spaces, spanning heterogeneous platforms. The basic contract between an object and its users is the interface that the object provides and users can use. Objects can be spread across the network for reasons of efficiency, availability of data, etc. From the perspective of the requester of a service, invocation should be the same no matter where the object physically resides.

The distributed object framework will be implemented using OODCE. The set of core DCE services are naming, security, threads, time and RPC. In order to aid the application programmer, another layer of abstractions is provided with OODCE. Four generic classes: DCEObj, DCEInterface, DCEInterfaceMgr, and ESO will be available for application programmers to implement client-server applications.

Electronic Mail (E-Mail)

E-mail is a standard component of Internet systems. It is useful for asynchronous, relatively slow notification of many different types. Also, E-mail is persistent, and will continue to try to deliver even if there are temporary network outages. The CsEmMailRelA class provides object-oriented application program interface (API) to send e-mail messages.

File Access ftp & kftp

The file access service provides functionality for file transfers and management. FTP is an internet standard application for file transfers. It allows a user to retrieve or send files from/to a remote server. The files transferred can be either ASCII or binary files. FTP also provides an insecure password protection scheme for authentication. KFTP builds on the standard FTP but adds a layer for strong Kerberos authentication. The CsFtFTPRelA object provides an object for managing FTP sessions between clients and servers to allow programmers to transfer files between machines. For Release A of ECS this service was built on the underlying TCP/IP network protocol using COTS. For Release B, the service will be provided over TCP/IP or HiPPI, using the same encapsulating object.

Remote File Access - DFS

Remote File Access (RFA) refers to the ability to mount remote files and access them just like local files. Distributed File System (DFS) provides location independent file access with high performance, availability and security. With the remote file access (RFA) capabilities provided by DFS, users can access remote files as if they are on the local file system. The whole process is transparent - upon authentication at login with the file servers up, no difference is noticed between a DFS file and files on local disk.

Interoperability with HiPPI: Bulk Data Service (BDS)

HiPPI (High Performance Parallel Interface) is a 800 Mbps high speed network technology. ECS currently plans to provide a HiPPI fabric that interconnects Data Server and Processing subsystem hosts to handle the large volume data traffic between the two subsystems. The HiPPI implementation is such that upper layer applications access HiPPI channels directly, bypassing the IP layer ("raw" HiPPI implementation). CSS will be using vendor provided HiPPI APIs to provide upper layer applications with a common interface, shielding the applications from the underlying HiPPI fabric. ECS has begun an analysis and prototyping effort to study the impact of implementing a HiPPI solution based on both IP-over-HiPPI and direct HiPPI channel access. If IP over HiPPI performance is at acceptable levels for ECS (several vendors are currently working to improve IP performance over HiPPI), then the interface with the upper layer applications that use the HiPPI fabric will be IP based. Please refer to a discussion on HiPPI in the ISS 305 document (305-CD-004-001).

Bulletin Board

Bulletin Board is another internet standard application, however unlike e-mail, Bulletin Board messages are directed to all readers of a named group. It uses the Network News Transfer protocol (NNTP) for sending and receiving messages. The CsBBMailRelA object provides object-oriented application program interface to send e-mail.

Virtual Terminal

Virtual Terminal access refers to remote logon to a machine. This software is provided on all platforms. The server telnetd will listen to incoming telnet clients and will allow remote logons. There is also a secure version of telnet and telnetd using Kerberos authentication which CSS will provide where available. This service is allowed only within ECS due to security considerations.

X-Windows is a Graphic User Interface conforming to the X/Open standard. While X is not a specific CSS Release B service this description is listed here for informational purposes. It consists of a client and a server where the client displays the actual interface. Developing applications in X is cumbersome and complex. OSF Motif is another standard, layered on top of X which provides a high level application programming interface to make the application development easier. Applications developed with Motif will work with an X server. The X client/server connection presents some significant security risks; therefore ECS will not generally support applications where the X client and the X server reside on different platforms. Users can down load data from ECS and can use the X application to view the data on their local machines. Alternatively, the program may consider providing dedicated circuit access or other means of securing X11 communication from a user client to an ECS X application.

ECS will provide user dial-up service from dumb terminal or PCs running terminal emulator. The dial-up service allows for the access to Internet and the ECS search and order tool character-based user interface (CHUI) for ECS users. Once a user accesses the search and order tool at a given DAAC, information search at other DAACs can also take place.

Event Log

Event log provides the programmers the capability to record events in to files. Events are broadly classified into two categories: management events and application events. Each event is recorded with all the relevant information for identifying and for later processing. Management events need to be recorded in a history file and on some occasions reported to the Network Node Manager. Application events are only recorded into a programmer specified file. Event log provides a uniform way for the application programmers to generate and report (record) events.

Server Request Framework (SRF)

The Server Request Framework (SRF) was designed to address the needs of ECS applications which handle asynchronous long-running requests. Many ECS servers (such as the Science Data Server) have this characteristic. The SRF infrastructure standardizes these aspects of asynchronous communication.

The infrastructure is intended for use by the developers of server applications. It will create and track service requests. It accepts call backs (e.g., event notifications) and sends them to the corresponding client object. It automatically tracks objects which can handle incoming requests or accept callbacks and dispatches the requests or callbacks to the correct object, and matches up synchronous requests with their responses. Client applications can send requests to server objects identified by their UR - the infrastructure will resolve the UR to the correct server object. Finally, the infrastructure provides factories for creating server objects (and the client counter parts) dynamically.

Subscription Server

The role of the Subscription Server is to support the detection of previously defined events and to perform specified actions on behalf of clients who have previously registered to those events. In order to actually perform its role in the ECS, the SubscriptionServer Process requires that its clients be active and have previously registered events with it. Examples of events include science granule insertion, metadata update, new advertisement, new schema export to DDICT, etc.

Process Framework (PF)

The ECS contains several infrastructure features which facilitate the implementation of client-server applications. The framework provides an extensible mechanism for ECS Client and Server applications to transparently include these infrastructure features. Therefore, its importance grows with future releases of ECS. Furthermore, the framework is used solely by ECS custom developed applications and as such is not meant for COTS applications.The primary objective of the PF is to ensure design and implementation consistency for all ECS Client and Server applications. This is achieved by encapsulating the implementation details of ECS infrastructure services and removing the need for programmers to rewrite common initialization code.

Universal References (URs)

Universal References (URs) provide applications and users a system wide mechanism for referencing ECS data and service objects. Once a UR is made for an object, the object can be disposed of and later reconstituted from the UR. URs can refer to objects that may be local to an address space, or remote. URs have a low cost to keep in memory and can be externalized into an ASCII string that an end user can manage. A UR has the capability of re-accessing and/or reconstituting the object into memory as needed. Therefore, the object does not have to remain in memory, and can if appropriate, be written to a secondary storage system, like a database.

3.4.2 HWCI Summary

The Distributed Communications Hardware CI (CSS-DCHCI) logically includes an enterprise communications server, a local communications server, and a bulletin board server. To provide for warm standby, the CSS servers and MSS servers at all DAAC sites and the SMC are cross-strapped and are configured to include the CSS Distributed Computing Software CI (including both OODCE client and server); the MSS Management Software CI; and the MSS Agent Software CI. This section only describes analysis of the CSS requirements. The complete configuration of the CSS and MSS HWCIs, based on the combined requirements of the subsystems and site-specific requirements, are presented in the site-specific volumes. Additional detail on the analysis of MSS HWCI sizing and performance is contained in the MSS volume. See Figure 3.4.2-1 and 3.4.2-2.

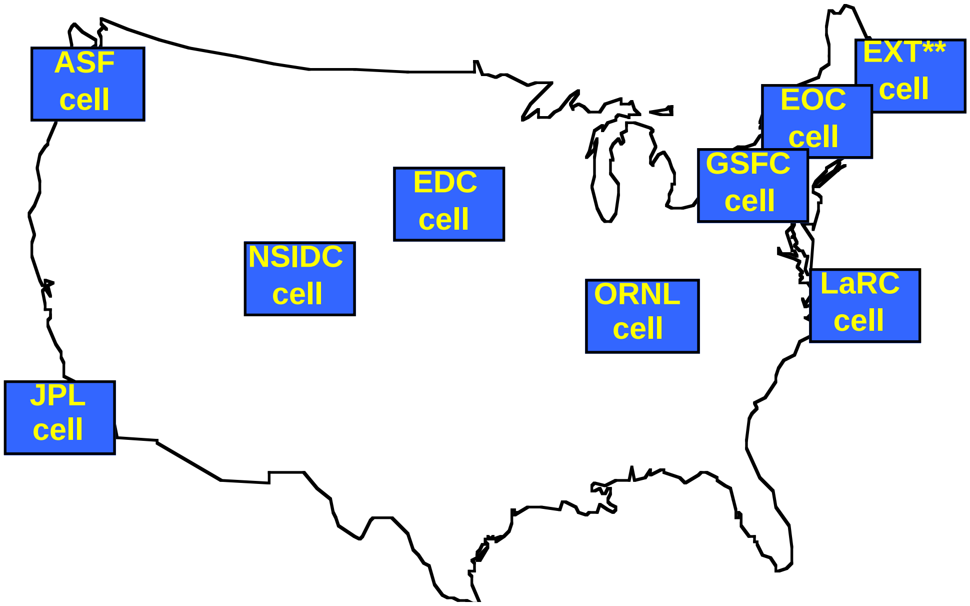
3.5 Security Implementation Overview

Security consists of hosts security and wire security. Hosts security is provided using native Operating System and Wire security is provided through DCE security feature.

The CSS, MSS and ISS subsystems all contribute to the ECS security implementation. ISS provides router-based security based on TCP/IP packet [IP address and port] address filtering. CSS provides Kerberos-based authentication, integrated authorization based on DCE authenticated RPCs and DCE access control, and data integrity based on encrypted checksums (provided by DCE). MSS provides virus checking, accountability, event reporting and analysis, and security policy coordination. Table 3.5–1 presents ECS security components and their contributions to overall security requirements. The CSS addresses the first four needs (refer to Section 4.2.2 for a detailed description of the CSS Security Service).

In addition to service-based contributions to security implementation, ECS has an integrated logical and physical security partitioning strategy involving DCE cell partitioning and use of isolation LANs at each of the ECS sites. Thus, although CSS encapsulates DCE services into higher-level, more abstract services for application developers, DCE plays an important role in maintaining the integrity of the entire ECS.

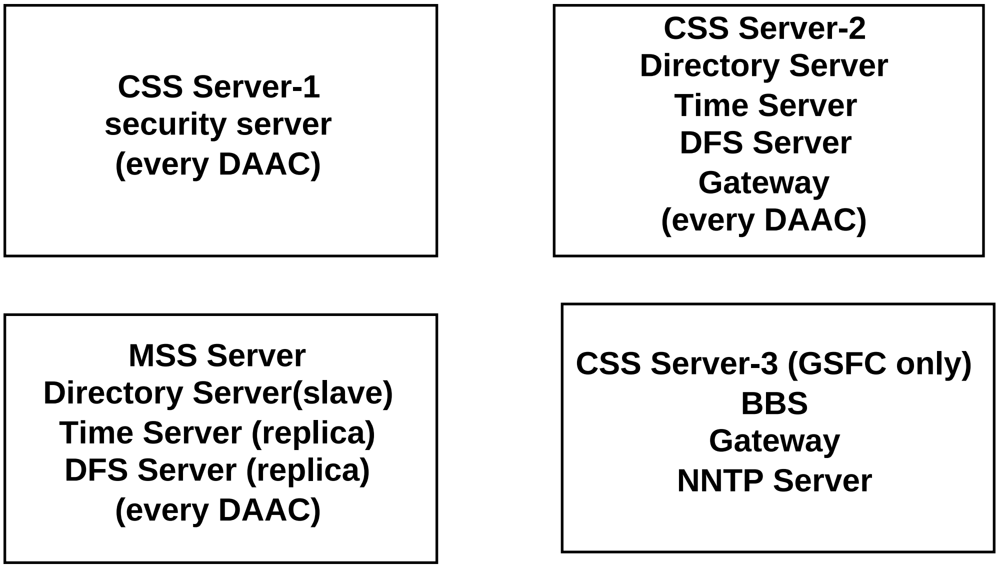
DCE Cells for Release B



** for all the external interfaces and gateways !! under discussion

Figure 3.4.2 -1. Release B Cells

Hardware Mapping for IDG



CSS Server-1
security server
(every DAAC)

CSS Server-2
Directory Server
Time Server
DFS Server
Gateway
(every DAAC)

MSS Server
Directory Server(slave)
Time Server (replica)
DFS Server (replica)
(every DAAC)

CSS Server-3 (GSFC only)
BBS
Gateway
NNTP Server

Figure 3.4.2 -2. Hardware Software Mapping

Table 3.5-1. Components of CSMS Security Implementation

Security Need	CSMS Security Implementation
Authentication	DCE-based Kerberos. Kerberized ftp, Kerberized telnet.
Authorization and access control	DCE access control. Router-based filters (port/socket at transport layer, and source and/or destination address at network layer). DCE cell configuration / “iso-cell” partitioning.
Data integrity	DCE-based RPCs (encrypted checksums).
Data confidentiality	DCE-based RPCs (encrypted data).
Countermeasures for degradation in network or processing resource performance through denial of service attack	Router-based filters; Support for nsu-IP-routing, application-layer gateways and bastions.
Security database management	DCE ACL managers, registry database.
Compliance management	MSS COTS & public domain tools for password audits, file system integrity checking.
Intrusion detection	COTS for detecting viruses, worms, Trojan horses, public domain tools (e.g, TCP Wrapper).
Security reporting	RDBMS.

3.6 Trades and Analyses Supporting Design

CSS design derives from a number of trade studies and analyses. These studies are diverse in scope, ranging from analyses of Ir1 design issues to analysis of evolvability and migration considerations in the choice of COTS products. Table 3.6-1 contains a summary of trades and analyses performed by CSS.

3.7 Changes Since Release A

This section addresses the major CSS design changes since Release A. These design changes have occurred due to:

- additional information being available from the major platform vendors, software COTS vendors, and technology consortiums;
- additional design detail is now available from the other ECS subsystems so that more accurate usage/need of CSS services is understood; and
- additional prototyping of CSS services and evaluations of potential COTS have been completed.

The major Release B changes include the use of DCE version 1.1, DCE cell architecture, -Remote File Access, Event Service, and Message Passing service. These changes are detailed in the following sections.

Table 3.6-1. Summary of CDR Trades and Analyses

Analysis	Objective	Conclusion	Future Analysis
DCE Encapsulation Prototype	Determine method for DCE encapsulation.	CSS recommends OODCE as the DCE Encapsulation Technique through Release B.	Track maturation of CORBA 2.0 products.
DCE Cell Configuration	Evaluate several cell configurations and their applicability to the needs of ECS.	Cell-per-DAAC configuration along with an external Isolation Cell (ISO+) was recommended at PDR. Cell-per-DAAC provides better security, autonomy of DAACs, and scalability. The ISO+ on top of the internal cell-per-DAAC provides a clear separation of external ECS users from internal ECS users providing better security. Implementing policy on these external users would be easier and more uniform than without an ISO+. Since all the needed features (cross-cell authentication, foreign identities support) to support multi-cell architecture are not supported in the available release of DCE, a single cell will be implemented for Release A and multiple cells for Release B.	Track the availability for Release B of Releases 1.1 and 1.2 of DCE that support multiple cell architectures.
Message Passing	Evaluate implementation options that satisfies the ECS subsystems requirements for asynchronous and deferred synchronous message passing.	The recommendation was to select a suitable COTS product that supports asynchronous message passing and custom implement deferred synchronous message passing, and other desirable features like security and callbacks. Several COTS products were evaluated and none could meet ECS requirements. We decided to develop this service on top of OODCE.	Track the maturation of CORBA 2.0 products, as it supports both asynchronous and deferred synchronous message passing.
Multicast Analysis	The main objectives of this multicast analysis are to: 1. Determine different methods of multicasting telemetry data on the FOS network. 2. Identify the potential for prototyping the solutions discovered in item (1).3. Analyze the impact of cost and policy issues pertaining to the proposed solutions.	CSS recommends implementing multicast by using Reliable Multicast Protocol (RMP) to provide reliable communications and C++ interfaces to the developers. CSS also recommends using Multicast Backbone (MBONE) technology and multicast routers that provide direct multicast support.	CSS further recommends that prototyping be done.
Remote File Access (RFA)	Analyze requirements for RFA, evaluate available alternatives (COTS) that provide the functionality and recommend implementation options.	DFS has a rich set of functionality. However, DFS did not fare well in the area of performance and robustness. Future releases of DCE are expected to include significant enhancements to improve performance. Rel B will use DFS in the areas of cross cell information/data.	Track DFS enhancements in future DCE Releases 1.1 and 1.2.

Table 3.7-1. Changes Since PDR and/or Release A

Changed Item	PDR Rel A Design	Rel B CDR Change	Notes
DCE Software Version	Rel A design based on using DCE Rel.1.1.	Rel A uses DCE 1.0.3, Rel B uses 1.1.	1.1 Not available from Vendors in time to be used in Rel A.
DCE Cell Structure	Rel A design: one cell per DAAC+ Iso-cell.	Rel A: one cell. Rel B one cell per DAAC + iso-cell. Rel B: multi cell.	1.1 needed for multi-cell with security - authorization and cross-cell authentication.
Remote File Access DCE DFS	Rel A uses DCE DFS for cross DAAC transfers. Uses DFS and ftp for User pull.	Rel A uses ftp for User pull. DFS available for prototyping/testing. Rel B uses DCE DFS	DCE DFS not robust with 1.0.3 and RFA not required with Rel A data volumes.
Event Service	Event Service used by MSS.	Rel A incorporates Event Service into Message Passing Services. Rel B evaluated event services, but will not support it separately either.	Event Service not being used by MSS. Decoupling of sender/receiver function not needed for Rel A or Rel B.
Message Passing	Rel A uses Message Passing via OODCE and Custom Code and Message Queuing COTS.	Rel A uses Message Passing and Queuing via OODCE and Custom Code. No changes for Rel B	Message Passing COTS are neither thread safe, nor reliable.
Bulletin Board	Rel A provides a standard nntp Server.	Rel A provides Web Server/ Browser. Rel B provides prototyping of secured web.	Web advancement faster than planned.

3.7.1 DCE Version and Cell Architecture Changes

The DCE cell configuration trade resulted in the decision to provide one cell per DAAC/site configuration, with an additional "Isolation cell" to isolate the above configurations from external networks. This trade, which is summarized in "Trade-off Studies Analytical Data" (ECS DID 211/ SE3), provides the basis for placement of an DCE/OODCE server at each ECS site. ECS architecture also includes maintaining (Directory & Security) replicas at each DAAC, SMC and the EOC to improve performance. The multi-cell architecture to provide scalability is planned to be implemented. Also, the new version of DCE (1.1) is planned to be used.

3.7.2 Remote File Access Changes

The addition of Distributed File Service (DFS) constitutes a major change in the Remote File Access facilities to be provided in Release B. DFS was not used in Release A because it was also felt that the number of users as well as the amount of data flow do not warrant DFS. Release B on the other hand will be implementing a multicell configuration and the number of users as well as the data flows are expected to be significantly higher. These factors have motivated the inclusion of DFS as a part of the Remote File Access facility. DFS has a rich set of functionality and provides complete security to the file level. DFS integrates well into the existing infrastructure (OODCE). Performance problems of DFS which were a cause for concern in Release A are expected to be mitigated in future releases of DCE (1.1 and 1.2) by including significant enhancements to improve the performance.

3.7.3 File Access Changes

Another important change related to File Access facility is the inclusion of a batch mode feature in the FTP service. Further, while FTP is a TCP application, the object framework by which subsystems invoke RFA will include an equivalent application implemented over HiPPI where applicable.

3.7.4 Event Services Changes

Event services are essentially the same as asynchronous message passing with the exception of the decoupling of senders and receivers. At PDR time frame, it was perceived that MSS would have to use this service to send notifications (traps) to the Network Node Manager. MSS design currently uses RPCs between the Agent and the NNM for reliable transmission of notifications. There hasn't been any other need identified in Release A or B for the Event Services. As such, Event Services is not being supported in Release A, and will not be supported in Release B.

3.7.5 Message Passing Changes

During PDR a trade study (“Trade-off Studies Analytical Data” ECS DID 211/SE3) was done to find the best way to provide Asynchronous Message Passing Service and it was determined that ECS would buy COTs products and develop custom features like Deferred Synchronous Message Passing on top of the COTs products. Since then, CSS has evaluated several COTs products and found that none of them are reliable or thread safe; They also have other limitations like message size and lack of security. It was decided at Release A CDR to custom develop this service on top of OODCE. Message passing will be developed on top of OODCE using threads and message queues. For further details about the design, please refer to Section 4.2.3.

3.7.6 Process Framework Changes

The Process Framework was designed since Release A CDR to address the need to ensure design and implementation consistency for all ECS Client and Server applications. This is achieved by encapsulating the implementation details of ECS infrastructure services and removing the need for programmers to rewrite common initialization code.

3.7.7 Server Request Framework Changes

The Server Request Framework (SRF) was designed since Release A CDR to address the needs of ECS applications which handle asynchronous long-running requests. Many ECS servers (such as the Science Data Server) have this characteristic.

3.7.8 Subscription Service Changes

The subscription server was designed since Release A CDR to address the need for a common mechanism across subsystems which can support a generic event-action model based on an event producer/consumer paradigm. Some of the users of the subscription server in Release B include Data Server, System Management, Data Management, Planning, Data Processing, Advertiser, etc.

3.7.9 Univeral References Changes

ECS functions are performed by manipulating logical entities represented at runtime as C++ objects in virtual memory. The Universal References Framework was designed since Release A CDR to address the need for a common mechanism across subsystems which can be used to reference logical entities beyond the time that it is computationally effective to keep the objects in memory. Universal References will be used by several subsystems in Release B including Data Server, Data Management, Planning, Data Processing, Advertiser, etc.